

SEG2105 Summer 2020

Assignment 5: Final Project

Jun Yi (Morris) Cai

S/N 300067686

Link to Github:

<https://github.com/MorrisJCai/ClientServerPandemicTestTracking/tree/PandemicTestTracker>

Video Walkthrough Link:

<https://youtu.be/jxKP3HLXQgg>

What I Made:

For this final assignment, I wanted to create something that both incorporates all the techniques and practices that were taught in this course as well as tackle the current circumstances regarding the COVID-19 Pandemic.

I created a pandemic test tracker system using the OCSF framework that was used in assignment 1, with the general objective and structure as the class diagrams from assignment 2. With this software, support workers both in the field and in the lab can communicate with each other about the performance of tests, the logistics of testing, and the result of the tests.

Requirements (Echoed on the README.md file on github):

- Client side must be able to specify if they are a test center, a lab center, or both
- Client side must be able to create tests with client information and a test identifier
- Client side, if it is a lab, must be able to edit tests that have previously been submitted to the server to mark them as positive or negative
- Server side must be able to connect to multiple test centers/lab centers at once
- Server side must be able to accept information from multiple test centers/lab centers at once
- Server side must be able to store and organize all incoming test data
- Server side must be able to display test information in a comprehensible way.
- All inputs must be able to be done with minimal dexterity (with gloves on or face masks)

User Stories

- At a testing center, a worker starts a new test. He/she grabs a test with a serial number, scans it, then inputs the patient information. After the test is performed, the server is notified of the test and the patient.
- At the lab, tests are received and tested. The result was negative and the information on the server must be updated to show that the test is negative.
- At the lab, a test is being updated but the serial number is scanned incorrectly/is damaged. The server replied that the serial number of the test is invalid and must be reviewed again.
- At the server/administrator center, a technician wishes to see the specific details about a test. He/she scans or inputs the serial number of the test, and the server displays the test information.

User Interface

As seen in the video, this software is designed to be used with a barcode scanner. It can both be launched by it (by scanning java ClientConsole and java ServerConsole), as well as interact with the interface by scanning commands.

This was designed for two reasons:

1. As mentioned in the requirements, this software must be able to be used when the user is wearing Personal Protective equipment (gloves, face masks, face shields, body suits, etc...) and they will have limited dexterity and motor skills. Using a barcode scanner instead of a keyboard and mouse will allow for workers to not fumble around with their limited mobility.
2. As a barcode scanner can be used without contact with surfaces, the only surface that has to be sanitized after use is the barcode scanner itself. For the patients being tested, they can easily hold up their health/insurance card and have it scanned without contact.
3. It allows for faster service and processing. As a barcode scanner can type text much faster than a human can, it lets long commands be typed almost instantly.

Build Script/Build Process.

As mentioned before, the build can be performed by scanning a barcode in an open terminal window. The user will have to open the terminal window manually though.

Class Diagrams:

Link to Umple Code: <http://cruise.eecs.uottawa.ca/umpleonline/umple.php?model=2007313fmrssiity>

